Compuverde
Software-defined storage

Compuverde

Technical Overview

Version 1.7.1.0
June 5, 2018

Abstract

This paper provides a detailed look at the architecture and components of the Compuverde storage system, functionalities, features and benefits. It covers the whole range of storage modes: Scale-out NAS, Hyperconverged, Metro and Async Replication – and goes into the details of the supported protocols and techniques that ensure the levels of compatibility, reliability, performance, scalability, and security required to meet the highest telecom grade standards.

Compuverde is an established provider of software-defined storage solutions for enterprises, service providers and telecommunications companies. Compuverde offers solutions that combine telecom-grade reliability and highly scalable, cloud-based object storage, enabling the use of environmentally friendly hardware which consumes less energy. Teamed with a top-ranked university and well-known partners within the telecom and IT industry, Compuverde is creating the future of storage solutions. You can read more at www.compuverde.com.

Contents

Introduction

Compuverde recognizes the necessity for enterprises to evolve from a fixed and rigid approach to storage, to one that is flexible and pragmatic. This is why Compuverde is delivering a fully software-defined storage solution that is hardware-agnostic and massively scalable, thus eliminating the cost and worry of future data migrations and hardware replacements.

Compuverde integrates a unified file system, object store and block storage into one package. The solution is defined as a cluster of nodes with a single file system spread across all nodes. Each node adds access points, cache, storage capacity and performance to the cluster.

Compuverde's architecture is balanced by definition, as all nodes play an equal role in the storage cluster. There is no master node, no metadata server, lock manager or dedicated gateway node. Users and applications can access the cluster through any node, spreading the load and efficiently eliminating bottlenecks. If your storage needs to grow or latency becomes an issue, new nodes can be added to the cluster and they will immediately take a load of the compute and storage effort. This ensures that what works today will continue to work in the future.

As all nodes are self-sustaining, self-balancing and true peers, the total performance increases linearly with every node added. The result is a high performing high-availability cluster with failover functionality for service and content.

There are four deployment modes to cover any use case:

- **Scale-out NAS**, known as **Compuverde vNAS**, is the preferred mode for maximum performance. In this mode the Compuverde software is installed directly on industry standard physical servers.
- **Hyperconverged** mode allows the Compuverde solution to be installed on virtual machines running on hypervisors, thus maximizing efficiency by bringing computing and storage together on the same physical machine (the hypervisor).
- **Metro Cluster** provides complete redundancy and creates a truly high-availability cluster for mission-critical data and applications. With Metro, your data is mirrored and distributed to two independent physical locations.
- **Async Replication** is a solution that connects two or more Compuverde remote sites and is very useful to achieve absolute disaster recovery, enabling possibilities of replicating selective data across sites.

Key benefits

- Ready to use "plug and play" scale-out storage for virtualization and cloud
- Extreme performance and scalability
- Reduced complexity
- No bottlenecks, hotspots or single point of failure
- Improved agility, lowered TCO
- One unified management screen
- Hardware agnostic
- Fully flash compliant
- Highly compatible with well-known protocols
- Telecom grade solution

Maintenance is an important part in any product's life-cycle, so it is worth noting that upgrades of the Compuverde software are done over the network, fully transparent to the clients, without service interruptions. During a rolling update, the system will take down one node at a time, using the virtual IP feature to pass the IP address temporarily to another node, do the update and then bring the node back online before proceeding with the next node.

System overview

Storage requirements today are often in the range of tera- or petabytes of data, and tend to grow exponentially. Compuverde's response to these needs is a unified file system on top of a cluster of storage nodes, to build a storage system that is hardware agnostic, scalable and easily configurable through one single management tool.

A simple setup could start with as few as four nodes, using existing or off-the-shelf hardware, then to be expanded as demands require.

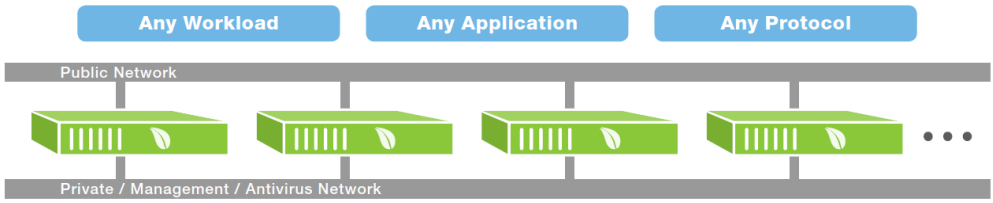

Figure 1: Compuverde cluster setup example

Note that, although the Compuverde software solution is hardware agnostic and can be built out of essentially any x86 based hardware, always refer to the system requirements list and updated reference architecture sheets when planning a storage solution intended for production, to ensure performance, availability, efficiency and feature support.

Nodes

A Compuverde node (also known as a storage node or a gateway node, as it will typically include both) is an x86 based computer with its own CPU, RAM, storage drives, and NVMe storage device for cache. The node can be "bare metal" or, for a hyper-converged solution, a virtual machine inside a hypervisor. The nodes provide ratios of capacity and throughput that add to the total of the cluster to which they belong.

Each node will perform better with a fast cache device, e.g. NVMe, as a buffer for data in motion, and simultaneously be able to continuously flush new data to the storage layer where typically many storage disks will share the load from a few cache devices. All-flash storage is supported for use cases where the main priority is performance rather than capacity. Given enough nodes, you can use tiering to combine both capacity demands and performance in one single storage cluster.

The Compuverde software is easily installed on each node. Everything is integrated so there is no need to pre-install an operating system.

Cluster

A Compuverde storage cluster is a collection of nodes that work together for common computing and storage purposes. One or more file systems span across all nodes in the cluster. All the nodes in the cluster have identical roles, which eliminates performance bottlenecks. Adding a new node to the cluster means adding its IOPS ability, cache and storage capacities to the total, resulting in reduced latency and a better user experience.

A storage cluster will be better balanced and performing, both on peak loads and over time, by having more nodes. More nodes mean that the system has more options when balancing the load and during the self-healing processes in case of hardware failure or if a node goes down. More nodes will also allow erasure coding modes providing higher redundancy, lower footprint, or both.

Network

Each node is equipped with one or more network interfaces. With one or more network switches, a number of separate logical networks are created to be used in the solution: public, management, private and antivirus networks. By using high speed, low latency switches, the throughput will be bound to cache and

disk speeds rather than I/O. Link aggregation and bonding is available for failover and added throughput when necessary.

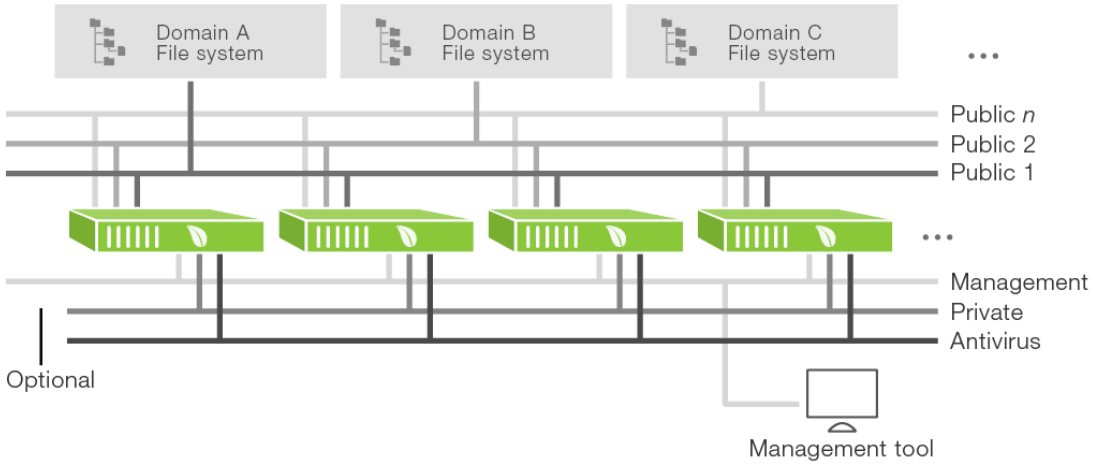

Figure 2: Compuverde cluster setup example with multiple logical networks

The storage nodes may use multicast messages, e.g. for heartbeats, to notify the entire cluster of its presence. For this reason, the network switches must support IGMP. All switches must have IGMP Snooping enabled, with only one switch in the group having both IGMP Querier and IGMP Snooping enabled. This must be done for each network separately, both the private and management networks. The storage cluster will use multicast messages only when direct communication from one node to another is not efficient. Multicast support is not required for the antivirus and public networks.

The public network is used for clients to access the storage, through any of the nodes, any of the available protocols, through a file system, object or block storage. You can have many public networks, one separate for each file system (multi-tenancy), or divide one single public network into subnets, one subnet for each file system. If so required, and at the same time, you can have multiple public networks to access one single file system. The public network is also used for communication with Active Directory, when joined. The main purpose of DNS on this network is for Active Directory. Keep in mind that the public networks are separated from the other networks described below.

In case of malfunction to one access point, the client will be seamlessly directed to another available access point by the use of Virtual IP. For more information, refer to the section Virtual IP.

The private network is used for all the nodes to communicate with each other, internally, and stay synchronized. Any change in one node is immediately available to another that requires the same content. The network is created automatically at setup and cannot be altered later. Make sure that this network provides the necessary bandwidth and with low latency. The private network must have IGMP enabled.

The management network provides a connection path between the management tool and the nodes. NTP, email, SNMP trap receiver and syslog server are also accessed through this network. The DNS on this network is used mainly for NTP lookup. A dedicated management network is recommended but not mandatory. As it is mainly used for gathering information and distributing configuration changes to the nodes when the administrator makes modifications, the requirements for latency and bandwidth are moderate for this network. If no dedicated management network is used, the management and private networks will be the same. The management network must have IGMP enabled.

The antivirus network is optional, in order to optimize the communication between the storage nodes and one or more external antivirus servers, when the antivirus feature is enabled.

Gateway cache

Gateway cache devices are used to dramatically reduce latency. The gateway caches throughout the cluster are always synchronized to ensure that every node has knowledge about which node owns a copy of a certain object so that a subsequent read or write can be handled by the appropriate node. This is much

more efficient than having to access the storage disks when content is not present in one node's cache but available in another.

RAM is the first level of cache. When the correct version of data is found in RAM, either from a previous write or a read, consecutive reads can be almost immediate. This is volatile storage, so write operations are not confirmed safe until data has been written to non-volatile SSD or disk. For persistent writes, make sure to not have synchronous mode disabled.

NVMe SSD or other fast, non-volatile storage acts as the next level of cache. Since this layer is much faster than traditional storage, latency is nearly eliminated. When possible, very small writes and updates will be accumulated before being stored to the cache layer. Changes can be further secured by using optional cache replication, which ensures that the changes are found on two or more nodes before being saved to persistent storage and replicated further.

As the gateway cache is shared throughout the cluster, it can be seen as having a cache pool with the total size of all node caches combined. For example, when a read command comes in to a node, the node can find that another node has the required data in its cache, ask for it and then deliver it to the client. This process is faster than if the first node would have to read the required data from spinning disks.

For gateway cache device, NVMe SSD is recommended. The use of SATA/SAS SSD gives significantly higher response time, lower throughput and longer execution time for operations like taking the gateway offline.

To avoid repeated writing onto the same cells of storage, the Compuverde software will aim to collect updates in blocks before writing to cache. Also, in a Compuverde solution, the same cache space is used for both reads and writes, reducing the required amount of operations in cases where, for instance, a file is first written and then read by the same or another client. Nonetheless, durability must be considered. Only use high endurance enterprise-grade NVMe SSD for cache devices.

Layered software approach

For performance and reliability, the Compuverde software is divided into three layers. The entry point to the system is the **protocol layer**, which ensures the communication between the client and the system through the protocol of choice (for example SMB, NFS or iSCSI). The **gateway layer** follows, where the main system logics reside, including the virtual file system and the cache layer. Next, the **storage layer** (object store) is where the data is actually replicated, distributed and stored. This layer only sees data as objects and has no information about file system or structure. The software layers are explained in more detail in the Software overview section.

Virtual file system

The Virtual File System (VFS) is the common file system created so that multiple clients can interact with the same stored data, regardless of which implemented protocols they are using. VFS resides in the gateway layer, above the cache.

The file system has been specially designed for scalability. It consists of small collections of metadata described in envelopes. Each envelope is associated with a folder and holds information about subfolders and files belonging to that folder. Instead of being centrally stored in a gateway node or a database, the envelopes are stored as objects in the storage layer, just as any regular data. This creates a robust architecture that allows nearly infinite expansion. When envelopes are stored in the storage layer, they are mirrored (3 copies for resilience 1) or erasure coded (2+2 for resilience 2) and stored on different nodes for protection, to ensure consensus and correctness in case of hardware failure.

Each file is organized in collections of data called extents, each extent being up to 1 GB in size, multiple to seamlessly form files beyond that size. Each extent has its own globally unique identifiers (GUID) and an offset value. The offset is used when data is added with an arbitrary gap from the previous part; having an offset value means that there is no need to store the gap itself. Each item in an envelope (i.e. item in a folder) has one or more GUIDs pointing to one or more extents. The envelope also contains information about item type, time stamps, snapshots, item size, its multicast IP address and more.

There is literally no physical limit to this structure, which allows for billions of files and folders. File names use UTF-8 for compatibility and each single file name and single folder name can be up to 260 characters in length.

The management tool

The Compuverde management tool can be installed on any Windows PC or VM connected to the storage cluster's management network. The tool is not part of the cluster or any node-to-node communication, it is merely a means to perform administrative tasks, such as configuring the cluster, monitoring the system, doing maintenance and accessing log files, all through a self-explanatory GUI.

Updated configuration settings are automatically distributed throughout the storage cluster. The management tool comes with a user guide that details each feature and option.

Product overview

The Compuverde solution can be set up in four distinct ways: Scale-out NAS, Hyperconverged, Metro Cluster and with Async Replication. Regardless of which you choose, the software installation and cluster setup are done in a similar few simple steps:

1. Install the Compuverde product software on each node and give a name to the node
2. Connect each node to one or more switches
3. Install the Compuverde management tool on a Windows client that can access the management network
4. In the management tool, run the wizard to create a cluster and add each node to it
5. Run the file system wizard to create a file system on the new cluster

See the Compuverde Quick Setup Guide for details when setting up the storage solution and the Management Manual for configuring authentication, multitenancy, file policies, snapshots etc.

Bare metal Installation

In this setup, each node is a self-sustained server with Compuverde software installed, complete with a set of predefined protocols. Each server is equipped with one or more gateway cache devices and a number of drives for storage. The figure below is a conceptual view of three of the nodes in a storage cluster. The system accepts interaction from applications and users through Ethernet and is keeping itself internally synchronized, horizontally throughout the cluster and vertically down to the storage. The file system is spanning over all nodes. Each node sees the same complete file system and acts as a file server towards the client accessing it.

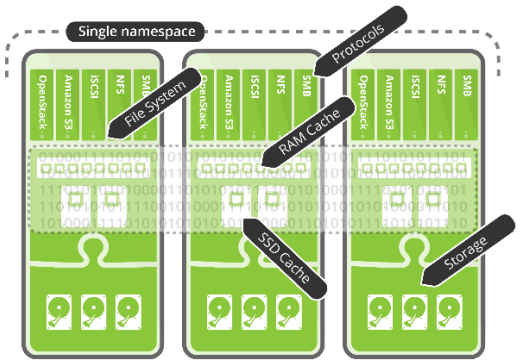

Figure 3: Schematic view of three nodes, Scale-out storage

Note: A Compuverde storage cluster starts with no less than four nodes. This is to allow headroom for self-healing and balancing processes to work properly. Having less than four nodes, and then losing one node due to failure, would set the system into a paused state and is therefore not recommended.

As a VM installation

Hyper-convergence takes all the benefits of server virtualization and extends them for both compute and storage. The need for fewer physical servers means that energy consumption is reduced significantly, in addition to saving valuable space. Without the need to make investments in expensive servers but instead expand “virtually” at a higher pace, yet at a lower cost, CAPEX needs and running OPEX are lowered.

In hyperconverged mode, the Compuverde storage cluster is built inside the virtualized environment by installing the firmware on virtual machines, one for each hypervisor host. Using passthrough or RDM, each node will access the cache layer and physical storage directly. This way, Compuverde provides a unified scale-out storage solution with file services that are accessible from any of the hypervisor hosts, both from inside for VMs within the hypervisor cluster and to the outside as a shared storage. Having the guest VMs, access points and file systems on the same hardware, has the potential of bringing the cache much closer to the client, reducing latency and speeding up read and write operations.

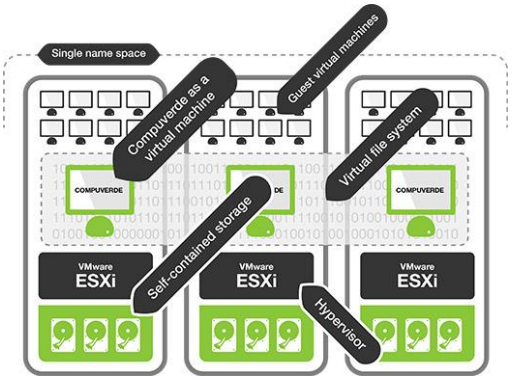

Figure 4: Schematic view of three nodes in hyper-converged setup

In this mode, it is recommended that each Compuverde node is provided PCI pass-through access to the host bus adapter where the disk devices used for Gateway cache and data storage are connected.

Metro Cluster

Compuverde Metro Cluster is an enterprise-grade, absolute high-availability solution designed to tolerate the loss of an entire data center, making it perfect for mission-critical applications that demand zero downtime.

Whether you deploy Compuverde NAS on bare-metal servers or on a hypervisor, Metro will mirror and synchronously distribute all data and changes to two independent physical locations. This gives you the ability to lose one entire location without losing data. The storage keeps itself internally synchronized, horizontally throughout the cluster and vertically down to the physical storage so that all nodes will have the same consistent view of the unified file system for all reads, writes and updates.

The two datacenters are classified as primary and secondary where the clients connect to the primary side. All nodes on both the primary and secondary side will have the same view of the files. All data changes and updates are mirrored and synchronously distributed to the secondary part. This gives you the ability to lose one entire location without losing data. Metro cache replication must be enabled for all writes and updates to be fully protected, including any write operations not yet flushed to storage. This will add latency similar to the round-trip delay between the sub-clusters as a result of writes being automatically mirrored to and confirmed by the secondary side. Leaving metro cache replication disabled will result in lower latency but at the risk of data loss should the *primary* side go down (i.e. the gateway state and writes not yet distributed to the storage layer on the secondary side).

Each location will be secured with its own erasure coding so that node failure will not affect the operation. Should the entire secondary side go down, the primary will continue to operate. Should the primary side go down, the secondary is still available but in order to continue operation, it will have to be temporarily promoted to become the primary part. This is to avoid any risk of “split brain”, should the connection between the two locations be lost.

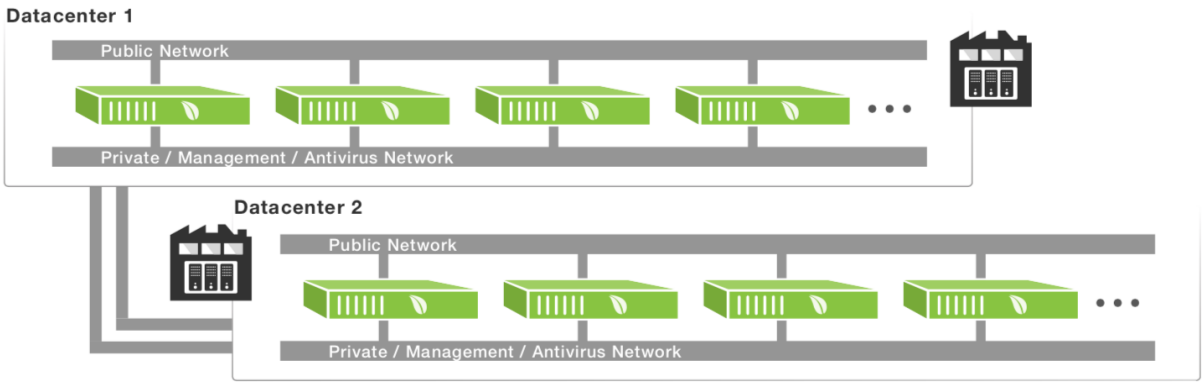

Figure 5: Metro Cluster

When planning a Metro Cluster, you can start with the primary side, just like any Compuverde storage cluster, except that the Metro feature should be activated. At any time, the secondary side can be added for full Metro functionality. The requirements for the connection between the two sites are 2 ms round-trip delay or better, and with a bandwidth equal to the private network. Multiple connections can be bonded for additional performance and redundancy.

Async Replication

Addressing the need for multi-location storage functionalities, Compuverde Async Replication combines the true flexibility of Compuverde Software-Defined Storage, with the ability to asynchronously replicate snapshots of data between multiple data centers, any distance apart, for the purpose of disaster recovery.

The Async Replication Disaster Recovery mode implies backing up the data stored on one datacenter to another. This way, if the source datacenter gets unavailable or damaged, its data can be completely recovered on the destination datacenter. Each datacenter can act as both source and destination for different shares.

An Async Replication environment contains up to 16 datacenters, which are located on different physical sites and act as peers. To be functional, the environment needs at least two datacenters. A datacenter is the equivalent of a domain / file system on a Compuverde cluster.

Each share is created on any of the datacenters and is mounted on any other datacenter. After setup, incremental snapshots are synchronized across the locations, over the internet, protected with TLS (Transport Layer Security) encryption. You can have different erasure encoding options on different locations, further enabling you to reduce data footprint and have added flexibility to your storage system. Up to 128 shares can be created in the Async Replication environment.

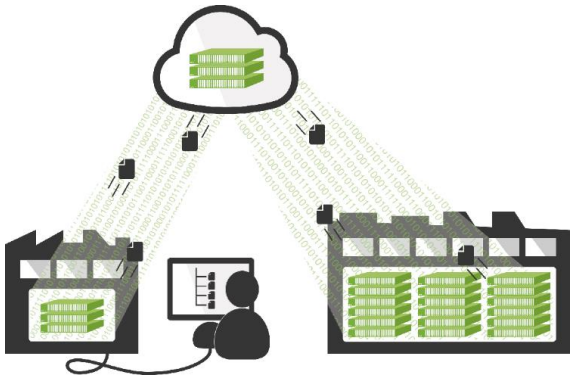

Figure 6: Async Replication

A Compuverde cluster is required on each physical site that is part of the Async Replication environment.

Quality attributes

Compatible

Compuverde provides access to the storage through various protocols, depending on the needs of the clients. If required, different clients using different protocols can access the same data simultaneously, on any gateway (access point), so that the choice of protocol is completely up to the client or the application. It is also possible to create multiple domains and multiple file systems within the same storage cluster, so called multi-tenancy – simplifying the administration and lowering the headroom requirements compared to having separate storage for each domain. Protocols include SMB, NFS. In addition, iSCSI, OpenStack Swift and Amazon S3 is available. For more details, please refer to the Supported protocols section.

The view of the file system through each node is strictly consistent, so that any modification on one gateway node is instantly available from any other gateway node. This is ensured by metadata being instantly synchronized, on cache level. A gateway node cannot deliver data unless it is confirmed to be the correct version.

Reliable

Reliability lies at the core of Compuverde, designed to provide telecom-grade availability of 99.999 percent or better, also called "five nines". This is achieved due to the symmetry of the architecture, no single point of failure, and by keeping the system core small, clean and efficient.

Compuverde uses erasure coding for data redundancy and protection. Data is striped across nodes and locations, not simply across disks as with traditional RAID. In case of hardware failure, each remaining node will be notified and start to recreate the missing replication data, while still being online. Since all remaining nodes are working in parallel to recreate the missing data, the data is quickly secured once again. Moreover, there is no wait time until data is available, it is kept available meanwhile being replicated in the background. This is crucial when dealing with large volumes that would otherwise require a long time to fully rebuild.

Virtual IP is a failover mechanism that makes all nodes in the storage cluster available at all times, even if one node should go down. With Virtual IP, one or many nodes will automatically acquire the one or many public IP addresses of the failing node and take the responsibility of continuing any data operations, until the failing node can be serviced and brought back online. Combine this with cache replication, a feature that writes to multiple nodes' caches synchronously, to ensure that all write operations continue on the other healthy nodes.

The number of nodes allowed to fail and still providing service, depends on the current resilience level and total number of nodes in the cluster. With five nodes or more and resilience level 2, you can lose any two nodes without interruption. Having only four nodes, with the same level of resilience, you can still lose any two nodes without losing data, but the two remaining nodes will not be able to provide service until at least one additional node is brought online.

Virtual IP is also used when performing rolling updates, enabling you to update the cluster system software on the fly, automatically and without downtime, totally seamless from the view of the client. The system will hand over all public IP addresses for one node to one or several other nodes and then take the node offline for update and reboot, one at a time. This way, the rest of the cluster will continue to serve.

Compuverde solves and avoids any "split-brain" and data inconsistency issues by immediately setting the cluster into a *No Cluster Agreement* state and stop operation to avoid damage if parts of the storage cluster become unavailable. The system will always look for majority, "quorum consensus", for instance, two out of three copies of metadata and two out of three slices of data in order to make changes. To avoid downtime, consider to add redundancy through link aggregation by using multiple network interfaces and switches.

High performing

Scaling out by adding more storage nodes provides a linear increase in average performance in the cluster. All the resources in the storage cluster are aggregated, like CPU, bandwidth, storage and cache pool, giving a persistent high throughput for a high number of simultaneous users.

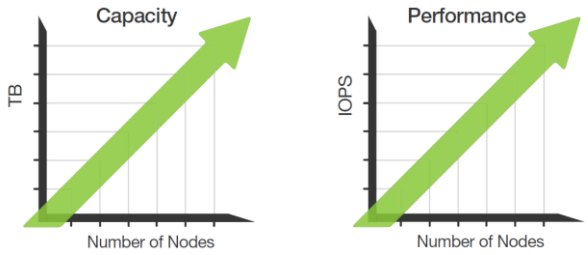

Figure 7: Compuverde Linear Scalability

Because all the nodes in the cluster also share the work of recreating lost data, the time frame from a hardware failure until the cluster is fully restored is dramatically shortened, with minimal impact on the system performance, data being accessible during recreation. The more nodes in the cluster, the less time it takes to complete the tasks.

Scalable

Every element of the architecture is designed for scalability – for instance, the metadata needed for the file system. Because metadata is crucial information that should be protected from loss, it is best stored as objects within the storage cluster rather than in the gateways, as seen elsewhere. Internally, each file and each block of data is identified with a 128-bit GUID. The metadata is cached, synchronized, duplicated and stored using the same algorithms as normal data. This not only creates a more robust and less complex architecture, but it also allows for extreme scalability.

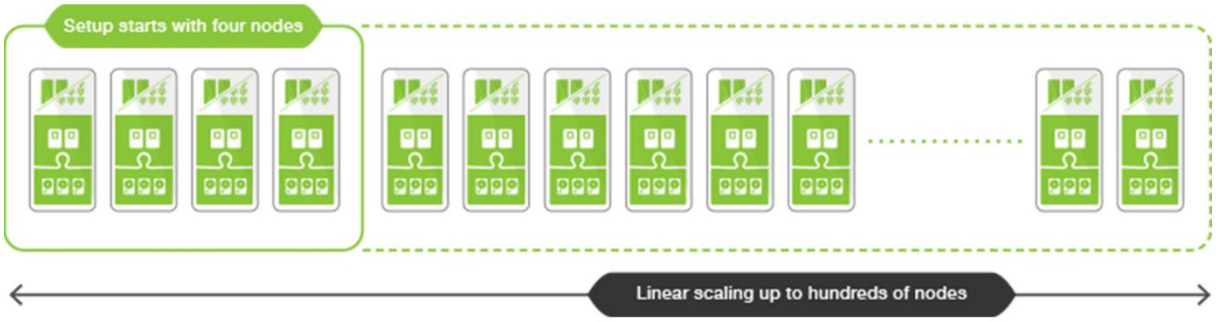

Figure 8: Compuverde Linear Scalability

There is no built-in limitation in terms of number of nodes in a single cluster. Tens or hundreds of storage nodes can be added to the same cluster, allowing petabytes of data. The cluster can be divided into tiers, having extremely fast all-flash nodes for performance and other nodes specialized on capacity for storing colder data, or for archiving. Still, clients are allowed to connect to any of the nodes, as the tiering is handled in the background, below the cache layer.

Secure

Data security is provided through authentication, as a layer of security can be introduced by verifying the credentials of the users or applications before allowing access to data. This includes Kerberos based authentication services, as well as Active Directory user and group lookup, NIS and LDAP.

Read more about authentication in the section Authentication.

Features

Multitenancy

Compuverde allows multiple, separate domains and file systems spanning over one single storage cluster, reducing costs by simplifying the administration and allowing available storage resources to be used more efficiently – as opposed to having separate storage for each domain. Each domain and file system are separated, using their own IP addresses for access, authentication mechanisms and set of protocols, still located on the same hardware. This way, the costs can be reduced as the overhead needed for the storage solution, in terms of CPU, infrastructure and storage capacity, can be utilized for all the domains combined.

Each file system is available through minimum one public network, using one public IP range. There can be up to eight public networks for each file system. This is useful when clients need to access the same file system and they reside on separate networks.

Disk quota

A disk quota limit restricts the use of storage space for a specific domain, file system, folder or sub-folder. The limit is specified in number of GB, TB or PB and will then apply for all users with access to the folder or share. The quota does not allocate storage, which would require an amount of overhead and less than optimal use of resources in cases where each quota is not fully utilized. This means that, if the administrator chooses, the sum of all the quotas on one hierarchical level may exceed the quota or limit for the parent level or the storage itself.

When accessing the file system, the share will be shown with the size of the quota or the size of available storage, whichever is smaller. If a write operation tries to write beyond the given limit, the operation will be aborted and the client will be notified that the write was not successful.

Virtual IP

Virtual IP is a failover mechanism to make sure that all the nodes in the cluster appear available at all times, continuing to deliver services when a node is taken down for upgrade or in the event of failure. This is done by moving the public IP addresses of the node going offline to one or more other nodes, which become responsible for these IP addresses in addition to its own. The result is that all IP addresses are available and the cluster continues to operate with the remaining nodes. The process is automatic and fast. The cost is a temporary decrease in performance until the affected node is ready to step back in and ease the load.

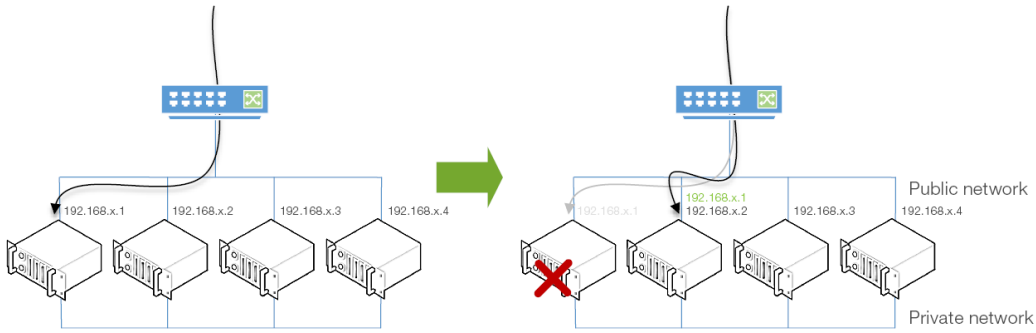

Figure 9: Virtual IP acquired by another node

When a node is taken offline in a controlled manner, the node will select other nodes in the cluster to take over its IP addresses and the transfer will be done transparently and seamlessly to the clients.

When a node goes down due to a failure, the cluster will detect the failure and then assign the missing IP addresses to other nodes. Note that if clients reside behind a router or other NAT devices, i.e. on a different

subnet than the public network they are accessing, the process may be delayed due to ARP information being updated by those network devices.

Auto rebalancing

The nodes in the storage cluster always aim to stay balanced, i.e. to utilize an equal amount of storage and IOPS compared to each node's total capabilities. Thus, when you add a new blank node to the cluster, the node would have to take a disproportionately heavy load of new data written to it. The Auto Rebalancing feature remedies this by moving existing data to or from any node found to be out of balance, thus effectively avoiding upcoming hot spots. As soon as the new node is included in the cluster, the pre-existing nodes are redistributing parts of their stored data to the new node. The feature will generate some additional data traffic on the private network and therefore it is not enabled by default. Enable this feature when adding a new node to an existing cluster.

This is an essential measure because the new node could otherwise end up being the only node to receive new data, which – as new data might have a higher probability of being read back soon compared to older data – would lead to a hotspot. The cluster stays balanced and all new data will continue to be evenly distributed.

Rolling upgrade

Compuverde supports rolling upgrades over the network so that the software can be updated or upgraded without taking the system offline. This is done through the management tool. By using Virtual IP on the nodes, each node will be taken offline and other nodes will seamlessly take its place. After the update, the node will join the cluster and reclaim its IP addresses so that the next node can be taken offline for update. This way, the cluster continues to deliver services uninterrupted during the entire update process. Please refer to the current Compuverde Software Update Guide before starting an update.

File policy

File policy is a feature to make automatic actions to files and content, for instance when they reach a certain age. It could be to move data to another storage tier, to add encryption to stored data, to set another file encoding, to set WORM or to simply remove old data. Using the management tool, a file policy can be set to a folder at any hierarchical level and will then apply to all files in all sub-folders below.

A file policy can be triggered by the age of the file (i.e. the time interval since the file was last modified: days, weeks, months or years). The files are chosen by file name pattern (e.g. "*.jpg", single and multiple character wildcards are allowed). When no age is specified, the rules apply continuously. A new file encoding (e.g. erasure 2+2) can be specified for files that are moved or re-written. If the intention is to erase files after a set time interval, since the file was last read or last modified, an optional retention period can be set. Another option is to have the files made WORM (Write Once Read Many) for archival. The files will then be set read-only and cannot be edited or deleted until changed to allow write.

Note that the file policy feature does not move or change the location of items within the file system. Instead, it can move or rewrite the data to another physical location on the storage (e.g. to another tier).

Snapshots

Snapshots give you the ability to retrieve earlier versions of files or folders in case of unwanted changes or deletion of files or folders. Snapshots are configured through snapshot policies, either by using the management tool or REST API, so that one snapshot can be made every hour, every day and once a week. Hence, with a limit of 253 snapshots in total for each snapshot policy, the weekly snapshots can be kept for longer than the daily and the hourly, until overwritten by new snapshots. For example, you can choose to keep the hourly snapshots for one or two days, daily snapshots for a number of days and weekly snapshots for weeks or months. Snapshots can also be taken manually, through the management tool or the REST API.

Logically, a snapshot is similar to a copy of the underlying file structure and all its files and content. Technically, to save space and to avoid loss in performance – and, more important, to ensure that all the folders and files in the snapshot are made at the exact same moment in time – the snapshot is merely a time stamp with information that the snapshot has been performed. The first time a change is being performed to a file after a snapshot has been taken, the part of the file that is going to be modified is copied before the file is actually modified (copy on write). Each update generates copies in block sizes as small as 128 kB. New updates within the same 128 kB block will still result, if no additional snapshot is taken in the meantime, in only one copy of the modified block.

The file system will seamlessly deliver files from each snapshot, when requested. As the snapshot is just a pointer to existing data and the data is stored on the same hardware, it should only be used as intended – never as a backup.

Once a snapshot policy is assigned to a folder, it will be valid for all sub-folders and files. Thus, there cannot be another snapshot policy on sub-folders. In order to assign a snapshot policy to a sub-folder, the first snapshot policy must be removed and then the new one applied on sub-folders.

Encryption

Data encryption is an optional feature that protects the confidentiality of data. Encryption is available both for data in motion and data at rest. When encryption is enabled for data at rest, data is encrypted before being stored in the storage layer. The encryption key is AES 256-bit with XTS and is unique for each file system. All encryption and decryption is CPU hardware accelerated by using AES-NI, AVX or SSE4.1 extended instruction sets.

Make sure that the option is enabled before data is being stored. Doing this later will not affect data already written, in which case you may want to move or re-write data by using the file policy feature in order to apply encryption.

For data in motion, TLS (Transport Layer Security) can be used – a cryptographic protocol that protects the confidentiality of data transferred to and from the Compuverde storage via Async Replication, REST API and web management. The TLS version currently supported is 1.2.

Antivirus support

The Compuverde storage cluster connects to one or more Symantec Protection Engine servers (SPE) to have antivirus scanning performed on all or selected parts of your file system, both in the background and live during reads. The antivirus servers should be connected through a separate antivirus network, alternatively through the management or private network. Communication between the storage cluster and Symantec Protection Engine server requires API 7.8 or later.

Antivirus scanning is controlled by antivirus policies defined on the file system, and it will be triggered when a file has been written to since the last scan or when the antivirus definition is more recent than the last completed scan. For NFS 4/4.1 and all versions of SMB, if a file is closed after writing, it will automatically be flagged for a new scan. The scan request will be sent from each node to one of the specified antivirus servers, either in the background or on the next read.

If a file is found to be infected, the file is flagged and cannot be read or copied; only written, updated, deleted or moved. Optionally, the infected file can be deleted automatically. Alerts can be sent to the administrator as email, SNMP traps or as messages to a syslog server. In case of a warning, a message can optionally be generated, not interrupting file operations.

Should an object be wrongly marked as infected due to a false positive, there are two options to unlock it: Either the scan must be performed again with an updated definition file where the mistake has been corrected, or the file can be moved to a folder without antivirus policy so that reading is not prevented.

Backup

The data stored in the Compuverde storage cluster can be backed up and restored by using the Network Data Management Protocol (NDMP) three-way backup. NDMP is an open standard protocol for network-based backup for network-attached storage. By using NDMP three-way backup, the backup server initiates the backup. Data travels over the public network by going from the storage cluster and directly to destination disk-based storage. Tape storage is not supported.

To prepare for a backup of your files, first create a snapshot policy on a selected folder. Onto this, create a backup policy. Then configure the backup server and start backup of the selected folder. By using a snapshot as the source of the backup, all the files remains consistent at the point of its creation.

Data protection

Data Integrity

Compuverde uses a variety of methods and techniques to ensure data integrity even in the most stressful working conditions, like heavy multi-concurrent IO activity to the storage, sudden power loss or hardware failure.

- Intelligent locking system allows multiple clients to do concurrent read and write to the same files at the same time, by protecting the data at the required level and byte range.
- No intermediate state in case of power loss or malfunction. Changes are confirmed only when stored on persistent storage (given appropriate configuration).
- Cache replication protects cached writes waiting to be flushed to storage layer (2-4 copies)
- No "split brain" condition in case of hardware failure or lost connection; Compuverde requires majority (quorum consensus) for changes to be performed.
- No hash collision due to truly unique 124-bit ID for each block of data, calculated from previous block ID.
- Self-healing storage cluster – all nodes in the cluster are responsible for monitoring, replicating and seamlessly recreating data.

Resilience

Resiliency is the ability of the storage cluster to recover quickly (or immediately) and continue to operate even when there has been equipment failure, power outage or other disruption to *parts* of the cluster. The resilience level of a Compuverde cluster, e.g. 2, determines how many nodes can be lost without losing data and without losing operation.

The exception is a four-node cluster when configured for resilience level 2. Because majority is required to ensure consistency, the four-node cluster cannot operate after losing two of its nodes, although the data is still intact.

When performing a rolling update of the nodes, one node will be taken offline at a time, temporarily reducing the resilience by one. Thus, for all storage clusters in production, to keep the resilience above zero at all times, the requirement should be resilience 2.

For resilience requirements above this level, for example to handle failure or power outage of an entire datacenter, then consider additional means to secure the storage cluster through added redundancy, for example redundant/dual UPS or through Compuverde Metro Cluster.

Erasure coding

To obtain the chosen resilience level, metadata and content are automatically duplicated to a minimum of three nodes. While mirroring is feasible for metadata and cached data, erasure coding is a more efficient method for the storage layer. While three copies of data, allowing any two nodes to fail, requires a footprint of 300 %, erasure coding 2+2, allowing any two nodes to fail, requires only two third of that: 200 %. Higher erasure coding, like 8+2, reduces the footprint to 125 %, giving a storage efficiency of 80 %, a massive improvement compared to mirroring.

The goals of erasure coding are to achieve storage efficiency, performance, reliability and availability of data. For resilience level 2, any erasure coding n+2 can be used, and for resilience 1, any n+1 (or n+2) is used. When losing one node due to equipment failure or other disruption, the resilience level will drop by one until the affected node is brought back online. Given sufficient headroom (free space), the cluster will take itself out of the vulnerable state by automatically copying missing pieces of data to existing free disk space. After the node is brought back online, any excessive pieces will be automatically removed to free up the space.

With erasure coding, for example 4+2, each block of data is divided into 4 slices, then 2 slices are added for redundancy. The total of six slices are distributed to any six nodes in the cluster (or within the defined tier), dynamically chosen for each block of data so that all the nodes in the cluster will receive pieces according to each node's individual capacity and location (e.g. tiering), eliminating hot spots entirely. This way, each node will hold a mix of both data slices and redundant slices from other node's data. In this example with 4+2, any four slices can be used to recreate the original block of data, meaning that any two slices can be lost without losing data. This means that any two disks or any 2 nodes can be lost. Or, when lost disks affect no more than (in this example) two nodes, any number of disks can be lost. The storage layer will seamlessly recreate the original data from any combination of remaining slices, while continuing to provide service.

Recommended minimum number of nodes in a storage cluster is n+2 e.g. for erasure coding 4+2, a minimum of six nodes should be used.

Table: Erasure overview

	Minimum number of nodes	*Nodes allowed to fail*	*Footprint*	*Storage efficiency*	*Note*
2+1	4	1	150 %	67 %	
3+1	5	1	133 %	75 %	
4+1	6	1	125 %	80 %	
5+1	7	1	120 %	83 %	
6+1	8	1	117 %	86 %	
8+1	10	1	113 %	89 %	*
2+2	4	2	200 %	50 %	
3+2	5	2	167 %	60 %	
4+2	6	2	150 %	67 %	
5+2	7	2	140 %	71 %	
6+2	8	2	133 %	75 %	
8+2	10	2	125 %	80 %	*

*) 8+1 and 8+2 not available for Metro Cluster

n+1 Similar to RAID 5, n+1 uses logical XOR and has a very low impact on the CPU, even for writes. Files are easily retrieved by using XOR "in reverse". It is a simple, fast and safe process.

n+2 Similar to RAID 6, there is a small penalty for write and update because the second slice is coded with XOR according to a table, estimated approximately four times as intensive as n+1. Reading a file is normally as fast, safe and simple as for n+1.

Note that even though the default erasure coding option is set for an entire file system, each block of data will hold information about its individual setting. So, if the option later is changed, then the existing files will still remain unaltered until the data is rewritten. Rewriting data can be done automatically by using file policies. File policies can also be used to decide different erasure coding for specific types of files or by age.

As erasure coding is applied in the background when data is flushed from cache to the storage layer, there is normally no additional latency introduced. For reading, even when hitting "outside" the cache, there is no significant latency because only four out of six pieces, or two out of four, is required (for 4+2 and 2+2 respectively). The storage layer is able to deliver the required data without waiting for all nodes and thereby keeping latency to a minimum. And with more disks being used, each disk will do less read time (in this example: one fourth each) to collectively read the same amount of data.

Cache replication ensures that, in case a node goes down, cached write operations not yet confirmed to be erasure coded and secured on the storage layer, will continue to be flushed (written to storage) in the background on another node. The default number of cache replicas is one, having all cached writes on two nodes simultaneously. Up to three replicas (four copies) are possible. Internally, all writes of all replicas are

confirmed before the originating write operation can be confirmed, which implies that, for performance, more replicas than necessary should be avoided.

Power loss

Regular storage operations, like storing and reading data, are secured by the balanced architecture with no single point of entry, i.e. no single point of failure, and by storage resilience and erasure coding, covered in the previous chapters Resilience and Erasure Coding.

In case of a power loss to an entire data center, what we do not want is the file system to be left in an intermediate and invalid state. This potential risk is eliminated by making sure that each change, both for metadata and data, is written to new blocks, and then only confirmed and included when complete. If, for any reason, a write should fail, e.g. due to power loss, then changes to the file system are automatically rolled back to its previous and valid state.

Secondly, the client should not receive confirmation until changes are complete and saved to persistent storage, e.g. non-volatile cache devices. For this reason, “sync” or synchronous mode is enabled by default, ensuring that the storage will not falsely confirm a write. When writes are not confirmed, e.g. due to power loss, the client should hold its data until able to write again. If synchronous mode is disabled, to gain performance, then there would be a risk of losing data upon power loss or other failure.

For protection against failure on cluster or datastore level, redundant/dual UPSs or Compuverde Metro Cluster would be necessary.

Self-healing storage cluster

A storage cluster can be in one of three possible states:

- Healthy cluster – All the storage nodes and disks in the cluster are online and operational
- Degraded cluster – One or more storage nodes or disks are offline
- No cluster agreement – The number of nodes online is not sufficient for normal functioning

A degraded cluster will start replication using available storage in order to secure the data.

A service, Cluster Monitor, is running in the storage layer of each node, monitoring its own node’s health and verifying through heartbeats that all other nodes in the cluster are alive. In the event of a hardware failure, power failure, NIC malfunction etc., the node is declared dead. Each and every node will automatically start replicating its own data (the parts that were located on the dead node) onto available storage throughout the cluster, thus keeping the vulnerability window to a minimum. Then, when a disk is replaced or a node is reinserted, no rebuilding of an entire volume will be necessary as would be the case for RAID disks, eliminating the chance of a second failure during a time-consuming rebuild.

Heartbeats make sure that a failing node will be noticed by all other nodes. There is no need for "hot spare parts". Instead, any missing pieces of data will be queued for replication and re-created by the other nodes in unison, using existing capacity and free space in the cluster. If alarm is enabled, the administrator will be alerted so that the broken node can be serviced and reinserted. If the node later comes alive, the now excessive pieces will be queued for removal, although with a lower priority. For each read, write, update or delete, the object store layer will look for issues and repair according to majority in order to keep consistency.

The system automatically checks the cluster and content in several ways to correct any potential error and to maintain security and effectiveness, including:

- Cluster Monitor / Heartbeats
- Search Cluster
- Read, Write and Update

Cluster monitor

Detection of lost nodes is done through heartbeats. A message is sent through the cluster and received by all nodes. When a node has not sent any such message for a predefined number of seconds, it is declared dead by the cluster. All the other nodes will start the Search Cluster operation in order to rebuild the content of the missing node.

Search cluster

When the cluster detects that a node is down due to missing heartbeats, each node simultaneously starts a local search for all files that were shared with the offline node, and immediately queue the results for replication using available storage on the cluster, ensuring that the data is still protected. This is possible because each node has information about all the locations of any data it possesses. When intentionally taking a node offline, a grace period can be set so that the cluster can hold back the replication activities for the given amount of time.

The replication process needs to be fast because we do not want to leave the system in a less protected state for too long. The process is extremely fast because each node is responsible for its own files, making it into a many-to-many replication with no central orchestration. Tests have shown that 24 moderately sized nodes were able to replicate 5 million unstructured objects, each of 1 MB in size, in only 19 minutes. With other replicating systems, a task like this would be measured in hours or days.

When the "declared dead" node later comes alive, after it has made itself available to the cluster, the replication process will abort and another process will start to remove the now-excessive (old) chunks of data. This is a separate queue with lower priority due to the fact that excessive data is less critical. Every chunk of data is equipped with a version number so that the latest or correct version is kept consistent.

Cluster health, status and replication can be monitored in real time through the management tool.

Read, write and update

Another part of the self-healing process is triggered by ordinary reads, writes and updates. Each time a file is accessed on the lower storage layer, a list with metadata is generated from all the nodes that store the file. All the lists from all the nodes are compared to ensure that all nodes are consistent. If any inconsistency is found, the issue will be queued for repair.

Queuing and prioritization

The self-healing processes use separate queues to prioritize the tasks that need to be performed in order to keep the storage cluster healthy and consistent. For instance, protection and replication are queued with higher prioritization than the removing of excessive data.

Authentication

Authentication services provide a layer of security by verifying users' credentials or applications before allowing access to read or modify data. Compuverde supports the following services, methods and protocols:

- AD - Active Directory
- LDAP - Lightweight Directory Access Protocol
- NIS - Network Information Service
- NTLMv1 / NTLMv2 - Windows NT LAN Manager
- Kerberos
- Local Users & Groups

Each node in the cluster automatically shares the same configuration, one configuration per file system, making it very easy to manage.

AD – Active Directory

Active Directory is a directory service for Windows domain networks. The Active Directory domain controller stores information about network resources and security principals (users and groups). Each security principal is assigned a unique security identifier (SID). The main reason for joining the cluster to a domain controller is to perform authentication.

Kerberos

Kerberos is a protocol for authentication. It is an integral part of Active Directory and is also used with other directory services. It provides enhanced authentication and standardization in order to cooperate with other operating systems. A challenge-response mechanism is used so that clients are able to prove their identities without sending a password to the server.

LDAP – Lightweight Directory Access Protocol

Lightweight Directory Access Protocol is an open, vendor-neutral, industry-standard protocol to enable access to directory services for authentication. Hence, LDAP can be used across many platforms. Active Directory uses LDAP for communication.

NIS – Network Information Service

Network Information Service is a directory services protocol. NIS is different from NIS+ which is not supported.

NTLM – NT LAN Manager - v1 - v2

Windows NT LAN Manager is a suite of Microsoft security protocols that provides challenge-response authentication, integrity, and confidentiality to clients. NTLM passthrough allows clients not joined to an AD to logon to a cluster that is joined to AD.

Local Users & Groups

An alternative to having a dedicated domain server is using Local Users & Groups that can be set up through the management tool. This is also where the optional secret key for Amazon S3 is specified.

Supported protocols

Compuverde supports client operating systems and clients using the following protocols:

- SMB
- NFS
- iSCSI
- OpenStack Swift
- Amazon S3
- NNTP backend storage

All data below the protocol layer, from the gateway with cache layer and down to the storage layer, is shared between the protocols so that changes made using one protocol on one gateway are instantly viewable to other systems using another protocol, even on another gateway.

SMB

SMB, Server Message Block, previously known as CIFS, is the network file system and directory service mainly used by Microsoft systems. Compuverde supports all versions of SMB up to 3.1.1 being used in Windows 8, Windows Server 2012 and later.

Supported features	
SMB 1	Basic features
	Home directory
	Authentication NTLM v1 - v2 and Kerberos
	ABE
	Offline folder
SMB 2-3	Crediting
	Server-side signing
	Notify: 1 level
	Oplock
	File leasing
	CopyChunk
	SMB Multichannel
	SMB Encryption
	Signing
	SMB Session

NFS

NFS, Network File System, is a distributed file system generally used by clients running UNIX and Linux. A client can access the file system the same way as if it were on a local file system without the client being aware of the actual location of the file. Compuverde supports NFS version 3, 4.0 and 4.1.

Supported features	
NFS 3	Basic features
NFS 4	ACL
	Authentication Kerberos
	Integrity Kerberos
	Encryption Kerberos
	File locking
	Session lease
	Client recovery
NFS 4.1	Exactly Once Semantics
	Session trunking
	pNFS File layout

iSCSI

One or more iSCSI Targets can be added to any cluster through the Compuverde management tool or REST API. One or more logical units, LUs (LUNs), can be added and connected to by any iSCSI initiator. By using Multipath I/O (MPIO), we deliver a high-quality and reliable storage service with failover and load balancing capability. Contrary to other protocols, due to the nature of iSCSI, the file structure will not be accessible or viewable through other protocols.

Supported features	
iSCSI	Multiple sessions from different initiators
	MPIO (Multipath) - The target can be reached on multiple paths (failover)
	Digests (CRC32 Error detection)
	CHAP Authentication

OpenStack Swift

OpenStack Swift is a RESTful Web protocol for a distributed object/blob store. Objects are stored in containers instead of files stored in directories, and they can be returned as XML, JSON or plain text. Data is organized using accounts, containers and objects.

Block storage is provided through an OpenStack Cinder plug-in. OpenStack Icehouse has to be installed, and there has to be an OpenStack controller node with Cinder Volume Service installed.

Supported features	
Swift	Accounts (get)
	Containers (get, put, delete, post, head)
	Objects (get, put, copy, delete, head, post)
Cinder	Operations on Volumes (create, delete, extend, clone)
	Create Image from Volume and Volume from Image
	Host assisted Volume Migration
	Manage existing Volume

Amazon S3

Amazon S3, Simple Storage Service, is an API for organizing objects in buckets. Buckets and objects can be created, listed and retrieved using a REST-style HTTP interface, returned as XML. You can also host static websites by using the website feature in Amazon S3.

Supported features	
Amazon S3	Common Headers
	Authenticating Requests - AWS Signature Version 2
	Operations on Buckets (list multipart, get, delete, location, head, put, acl)
	Operations on Objects (list, get, head, post, put, copy, upload part/multipart, delete, acl)
	Operations on Bucket Website (put, delete, get bucket website)

NNTP

Compuverde provides a limited command set to support NNTP backend storage, so that Usenet servers, e.g. Diablo Usenet Software, can connect to a Compuverde cluster for storage.

Supported features		
NNTP	Upload	IHAVE, TAKETHIS
	Download	ARTICLE, HEAD, BODY
	Info	CHECK, STAT
	Delete	DELETE
	Management	HELP, CAPABILITIES, QUIT, MODE STREAM

Software overview

To ensure performance, compatibility and reliability, the system software and all the protocols are implemented by Compuverde in C/C++. This is to ensure a small core, efficient memory allocation, and to have a much tighter and consistent integration between the protocols and the internal file system – so that all implemented features in one protocol can be made interoperable in real time with features in any other protocol. The result is a smaller footprint, stability, predictability and high performance.

For example, if one user opens a file in SMB and is granted a range lock on that file, and another user opens the same file in NFS, then the first range lock is still valid, even if the two connections are made through different protocols or different gateways.

There is no use of third-party or open source libraries, giving the developers full control of the code, resulting in a smaller footprint and better reliability.

Software layers

For reliability, compatibility and flexibility, the software architecture has been divided into three main logical layers:

1. Protocol layer
2. Gateway with cache layer
3. Storage layer

While the Compuverde storage node contains all three layers in one, this strictly layered approach also allows for installations where the gateway and storage are separated (protocol, gateway and cache, and storage layer, respectively). The gateway nodes then communicate with the storage layer through the private network. This way, each part can scale independently, as illustrated in the following three graphs. From left to right: Scaling by adding Compuverde integrated nodes, adding storage nodes (capacity) and adding gateway nodes (performance).

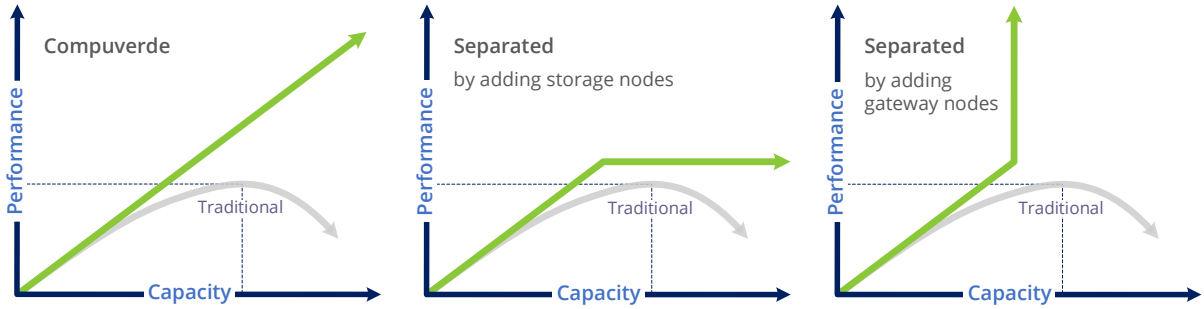

Figure 10: Compuverde scale-out storage compared to separated alternative 1 and 2

Protocol layer

The protocol layer is what makes Compuverde compatible with a wide range of existing systems, OSs and applications. Data below the protocol layer is shared between the protocols so that changes made using one protocol or file system on one gateway are instantly accessible to other systems using another protocol or file system, on the same or another gateway. However, iSCSI can be viewed as a "bucket" of storage, still stored in the storage layer the same way as other data, not accessible through any other protocols.

The protocol layer delivers the file system and functionality for SMB, NFS, iSCSI, OpenStack Swift and Amazon S3.

Gateway and cache layer

The gateway and cache layer consists of two main parts:

- Gateway with Virtual File System
- Cache

The gateway redirects and conducts all file operations, uses envelopes to describe the file system structure and communicates with all other gateways in the cluster. The gateway ensures that cache writes are mirrored, when enabled, and uses the cache devices for faster reads and writes. For the cache, there is a front-end and a back-end API. The gateway communicates with the cache front-end and the object store communicates with the cache back-end. The cache holds information about what parts of files need to be stored in the object store, denoted “cache dirty”. A background process running in the object store then handles these objects and makes sure they are replicated, erasure coded and stored on multiple nodes.

Information about all the caches are synchronized throughout the cluster so that any read or write can be redirected to the cache holding the file.

Storage layer

The storage layer handles data as objects and has no notion about file structure. It communicates with other nodes’ storage layers through the private network interface in order for the replication to work seamlessly.

The storage layer consists of four main parts:

- File Service: Localize, read, write and delete files
- Replication Service: Replication of missing parts and removal of excessive parts
- Cluster Monitor: Heartbeat and condition of all other nodes
- Management: An API for configuration and information

The File Service is responsible for saving all cached writes and to retrieve non-cached data, both of which will be available in cache for succeeding reads and writes. The Replication Service will take care of data redundancy. The Cluster Monitor is responsible for monitoring the cluster through heartbeats and to scan content for any inconsistency. The Management API provides information and means to update the configuration, for the management tool.

The self-healing and replicating processes ensure that data at all times is secured and that the gateway will have a consistent and clean access to the storage. At the same time, the available capacity is kept balanced.

Read operations

A read request starts with a certain protocol’s respective read command to the gateway of a node of choice. The node will quickly look up and determine whether the file is owned by another node. If the file is owned by another node, the first node will check if the file is found in its own cache, and, if found, ask the owner node directly whether this file can be used as it is (i.e., if the respective version is the correct, thus ensuring data consistency). If yes, then the read data can be returned immediately. Otherwise, the request is forwarded to the node that owns the file.

The node that owns the file will look for the content in its own cache, and then any other node's cache. If found, the node will return the content, in this case almost instantly.

If the data does not exist in the cache layer, the node will ask the storage layer to retrieve it. The storage layer will send a message asking for file slices. Each node that can provide a piece will reply directly to the node asking. For consistency, the version number is included, together with information about the node’s current load, in addition to a list of every other node known to share the same file. The node then decides what nodes to retrieve the file from. Pieces are collected, joined and returned. This whole process is seamless and invisible to the user.

When a client asks for a very small piece of data, the whole block where that piece is located is cached. If the client later wants to read more from the same block, the data is readily available in cache.

Write operations

A write or update request starts with a certain protocol’s respective command sent to the gateway of one node of choice. The gateway layer will determine whether the file that the write is addressed to is owned by

another node and, if so, the request is forwarded to that node. For protection, when cache replication is not disabled, data is sent to another node so that it will be written to two (default) or up to four caches at the same time, in case of failure to one node in the middle of writes.

Technical information: The content of each file is handled in blocks of up to 1 GB in size, named extents. Each extent is given a unique 128-bit GUID, calculated as a variation of the previous GUID in such a way that hash collision is not possible. In addition, each extent is given a version number and an offset value that indicates the start position of the extent inside a larger file. 124 of the 128 bits in the GUID are truly unique for each extent, giving 2^124 unique GUIDs, assuring that data can never be accidentally overwritten.

In the cache layer, each extent is split into 128 kB blocks identified by GUID, file block index and version number. For efficiency, when possible, very small updates are accumulated in RAM before they are saved to NVMe or SSD. When appropriate, or when updates are committed, space is allocated on the cache disk for the new parts, for each 128 kB block independently, and then written. The gateway returns with a confirmation that the file is successfully secured. In the cache layer, there is a list of block entries that tells the object store which blocks need to be saved to storage and with what type of redundancy.

Note that erasure coding is not used in the cache layer, but instead in the object store layer. Erasure coding in the cache layer would make write and update a more complex and time-consuming task, requiring the system to collect pieces, then update, recalculate and distribute the new erasure-coded pieces. Instead, the data belonging to write and update operations is mirrored to different caches for protection.

In the storage layer, each block that was scheduled for storage is sliced into a number of parts, according to the erasure coding redundancy level. For instance, for erasure coding 4+2, each slice will be 128 / 4 = 32 kB, referred to as the stripe unit size, plus two parts generated for redundancy, with same unit size. The storage layer then asks the cluster which nodes are available for storage. Each node in the cluster returns information about its own load and free space directly to the node responsible for the file. The six pieces are distributed accordingly, considering:

- Geographical position
- Free space in the storage node
- Current workload
- Tier policies
- Outcome of a random generator, to avoid any repetitive patterns and potential hotspots

The data pieces are now distributed and the storage layer is going to store the data physically to disk on six different nodes. When the process is successful, the cached file is marked as “clean” and is kept in the cache until new blocks of data require the space.

Multi-threaded I/O

With any vNAS, cloud storage and server virtualization, there is a need for high throughput and low latency for large files and multiple concurrent users. A Compuverde storage cluster can handle thousands of I/O requests simultaneously, in parallel, handled intelligently with byte-range file locking and a deep queue for waiting requests.

Per node:

- Parallel I/O operations: 16 + number of GB RAM
- I/O queue depth: 256 per client

The demand for I/O queues is usually lower both because the traffic is shared across the cluster, and due to the existence of the cache layer within each node. One or more NVMe devices are used per node to keep the queues down and latency to a minimum.

Locks and concurrency

Concurrent reading or writing of files is possible, even across platforms and between nodes. For instance, when a client opens a file for reading, the client will be granted a shared lock so that multiple clients can

read, not write, the same file on any gateway through any protocol. For writing, the client will request an exclusive lock to keep out other clients that may want to read or write to the same location at the same time. This can be done for a specific byte range so that multiple clients can simultaneously read and write to different parts of the same file. This is essential for OSs and virtual machines, as they often need to read and write to different parts of a file at the same time.

Example of simultaneous file writing and reading from different nodes, to the same file:

1. File read is initiated by two clients. The requests are redirected to the owner of the file
2. Shared lock is granted to the two clients and read is started in parallel
3. A third client requests an exclusive lock to write to the same file
4. The shared (read) locks cannot be reclaimed so the write will have to wait
5. When the reads are finished, exclusive lock is granted and the write can begin

If, instead, the reading clients only request locks for some part of the file, and the writing clients request to update another part of the same file, the write process would not have to wait.

Also supported are SMB opportunistic locks, or oplocks, and NFS delegations, so that clients can request files for caching locally. This allows for faster updates using less network traffic and improved response time. If the lock is possible (no other clients have the file open), the request is granted. When another client tries to access the file, the first client is notified so it can do appropriate actions, flush and close the file, before the new client can proceed. If the first client for some reason cannot answer, then the opportunistic lock will be degraded, thus invalidating the local copy of the non-answering client, so that the new client can access the file.

System requirements

Recommended system requirements for a Compuverde vNAS storage cluster, per node, for use with products in regular, integrated mode:

	Recommended	*Note*
CPU	Intel Xeon E5-2620, 4110 or similar	Minimum x86-64 (4+ cores) supporting SSE4.1, one or two CPUs depending on use case
RAM Memory	32+ GB	Minimum 16 GB
Boot disk	60+ GB	Minimum 20 GB
Cache disk	100+ GB NVMe (one or two)	Cache < 100 GB possible, not recommended
Storage disk	SAS/SATA disks (multiple disks)	
Controller card	HBA (IT mode)	
Network	10 Gigabit NIC or more	Multiple networks created by using VLAN
Number of nodes	Scale-out from 4 nodes	The cluster can run with 3 nodes
Network switch	10 Gigabit switches	Multiple switches for redundancy

IGMP is required for all switches that handle the cluster's private and management networks.

The recommendation for cache, to obtain optimal functionality and throughput, is NVMe, 100 GB or more. For durability, high endurance enterprise-grade NVMe or SSD (DWPD > 9) should be used.

The boot disk holds system log files in addition to OS and firmware. The recommended size is sufficient, as no more than a few GB of truncated log files are usable. Mirrored boot disks (RAID1) is an option when availability is highly important, to lower the risk of having to reinstall a node and reinitialize the data disks. Please contact Compuverde for information on RAID compatibility.

For production, to secure writes not yet stored to the storage layer, make sure that cache replication is enabled (default). In addition, there should be sufficient usable storage capacity available to allow the self-healing processes to recover data, should one or more nodes go down.

Resilience level is selected at initial configuration and cannot be altered later. If you plan on using the cluster in production, resilience 2 should be used. In combination with the Virtual IP feature, this allows for safe non-interruptive rolling updates. Four nodes are sufficient for a fully functional cluster. The recommended minimum number of nodes is determined by resilience and erasure coding to be used, equal to $n + k + 1$. Thus, for erasure coding 2+2, giving a resilience of 2, the recommended minimum number of nodes is five.

The amount of RAM required for each node depends on the number and capacity of disks installed on the node; with greater capacity, more RAM is useful, for buffering metadata, data, and to reduce latency.

The number of storage disks in each node will have a direct effect on the maximum sustainable I/O performance of the cluster. For example, having twice the amount of storage disks reduces the reads and writes for each disk by half and thus improves the total throughput.

For the storage disks, RAID is not recommended as, for this usage, it would add complexity without any significant advantages. Identifying a defective disk within a node would be difficult or time consuming, as the node will have to be booted into RAID BIOS or using tools not part of the Compuverde solution. If a decision is made to use RAID for storage disks, then single storage drives in RAID 0 volumes with RAID write-back feature enabled is an option. When using the write-back feature, make sure that a Battery Backup Unit is present to keep data safe. Instead of RAID, the recommendation is to use HBA for the storage devices, in IT mode only.

Summary

With data storage demands rising exponentially, service providers and large enterprises are searching for ways to leverage customer data efficiently and safely while controlling spiraling energy expenses.

The Compuverde unified storage solution offers file system and scalability simply by using x86 storage servers as building blocks, forming them into high-performance storage clouds. The use of a symmetric architecture eliminates bottlenecks that are a costly part of traditional storage solutions. No complicated central orchestration, no special purpose hardware, single points of failure or performance bottlenecks to protect. Everything is managed through one single user interface, making all operations effortless.

- **Simplified deployment and maintenance**: The Compuverde software architecture avoids the need for special-purpose or central nodes, greatly simplifying deployment and maintenance.
- **Faster than traditional storage and similar storage solutions:** All disks and all storage nodes work in parallel with no central orchestration and no costly bottlenecks.
- **Performance and reliability**: Data is stored in multiple copies and can be separated geographically among different locations within the cluster. A highly efficient fault recovery process allows for nearly complete redundancy.
- **Supporting all storage types**: The unified software-defined platform supports file, block and object storage to manage both structured and unstructured data.
- **Self-healing**: When a disk or storage node breaks down, the cluster self-heals by recreating the missing data, resulting only in a temporary capacity decrease.
- **Lower cost**: By leveraging clusters of standardized servers, Compuverde stores petabytes of accessible data at the lowest possible cost.
- **Pay-as-you-go:** The ability to start small and scale out on demand ensures efficient and optimized growth, without the need for migration or throwing out existing hardware. This makes the solution affordable and lowers total cost of ownership.

Technical specifications

System features

Storage modes	Scale-out NAS (bare metal) Hyperconverged Metro Storage Cluster Async Replication
Access Protocols	SMB (1 / 2.0 / 2.1 / 3) NFS (3 / 4.0 / 4.1) iSCSI (with MPIO) OpenStack Swift + Cinder Amazon S3 NNTP back end storage
All-Flash	Yes
Tiering	Yes
Multi-tenancy	Yes (multiple file systems)
Network	
Virtual IP	Yes
Multiple NICs	Yes
VLAN	Yes
Bonding	Round Robin Active/Backup XOR Broadcast LACP Adaptive Transmit Load Balancing Adaptive Load Balancing
Rolling upgrade	Yes
Cache support	NVMe / SSD / RAM
Cache mode	Read/Write / Write Only
Mirrored write cache	1-4 copies
RAM write cache	On / Off
Authentication	Active Directory LDAP Kerberos KDC NIS Local database

Management features

Management tool	Yes (Windows based) Performance Configuration Health Alerts
Alarms	SNMP E-mail
Logging	Yes, management tool Remote syslog (rsyslog) SNMP v2c
API	REST API Usage statistics
Backup	NDMP
OpenStack Cinder	Yes

File system features

File Policy	Yes (at folder level)
Filters	Pattern (ex: *.jpg)
	Age (days/weeks/months/years)
Actions	Change file coding
	Change tier
	Data encryption
	Retention (remove files after set time)
	WORM (write once read many)
Snapshot Policy	Yes (at folder level)
Schedule	Automated: Every hour, day and week
	Manual through REST API
Snapshots retained	Up to 253 (circular overwriting when threshold is reached)
Quota Policy	Yes (at folder level)
Folder size limit	GB / TB / PB
Antivirus support	Symantec Protection Engine
File coding	Copies
	Erasure coding
Copies	3 or 5 copies
Erasure coding	2+1, 3+1, 4+1, 5+1, 6+1, 8+1, 2+2, 3+2, 4+2, 5+2, 6+2, 8+2

Cluster features

Scalability	Linear, by adding new nodes
	Petabytes of data, billions of files
Elasticity	Runtime change of cluster size
Self-healing	Yes (Automatic)
Automatic detection	Node failure
	Disk failure
	Data inconsistency
Healing mode	Prioritized automatic repair
Availability	> 99.999 %
Data addressing	File system
	Block
	Objects
Encryption	Data at rest: AES 256-bit XTS, one key for each file system

Index